

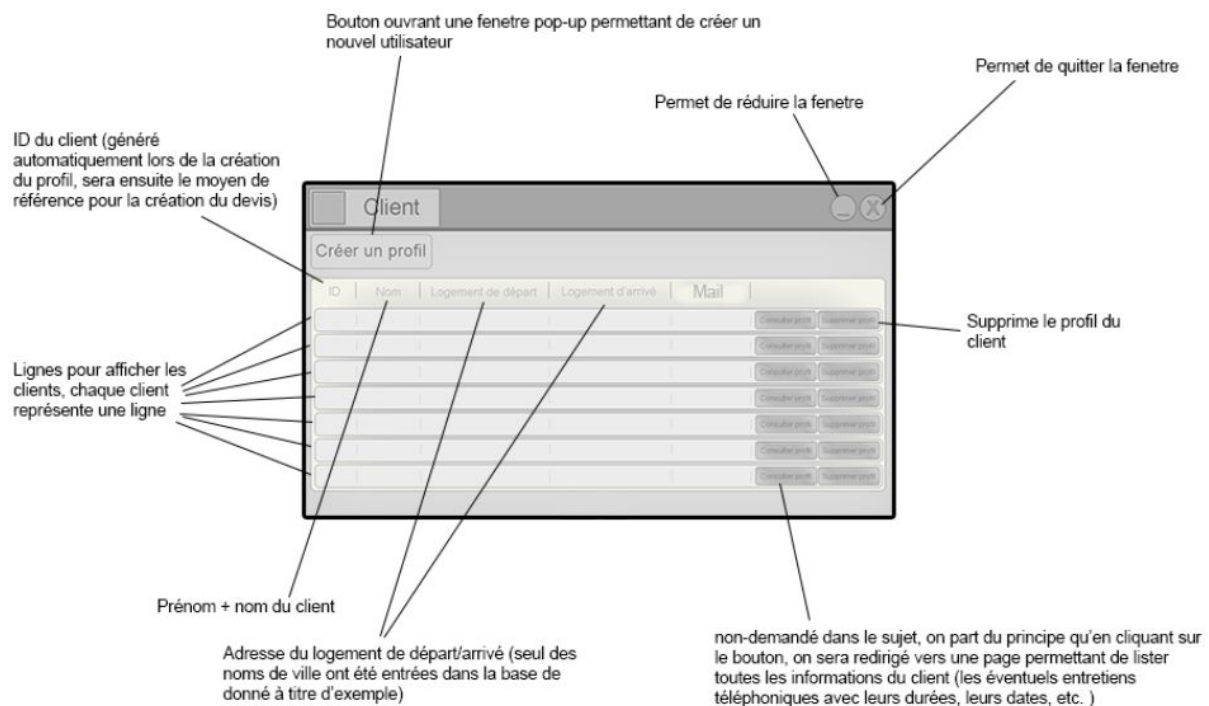
## Sujet 1.2:

Nous devons créer une application permettant à un commercial d'établir un devis pour ses clients. Nous estimons que la création du client n'est pas encore faite, puisqu'il n'est pas question de création de profil client dans le sujet 1.1, comme nous devons associer le devis à un client, nous avons besoin que les clients soient créés: le premier onglet de notre application permet donc de consulter tous les clients et d'en créer de nouveau au besoin. Le second onglet est l'onglet principal de l'application, c'est lui qui permet de créer des devis, le commercial rentre toutes les informations concernant le coût de transport, les informations sur les logements d'arrivés et de départ en prenant en compte les éventuels service que le client pourrait demander. Le dernier onglet permet de consulter le planning des camions, pour savoir quel camion est disponible à tel heure, quel camion est utilisé de telle heure à telle heure et savoir où il se trouve.

Graphiquement notre application est adaptée aux tablettes au format paysage, selon le sujet: les déménageurs travaillent avec une tablette et très peu de personne utilise une tablette en format portrait, on a donc estimé que le format le plus utilisé serait le paysage. Le thème ne change pas en fonction des menus, une fois l'application prise en main, il est difficile de se perdre, les choses essentielles sont condensées.

### Onglet Client:

Dans l'onglet client on retrouve la possibilité de créer un profil, de voir tous les clients déjà existant, de consulter leur profil ou bien de le supprimer.



Pour créer un utilisateur, les seules informations demandées sont: Le nom, le logement de départ, le logement d'arrivé et le mail du client; l'id est automatiquement généré et on part du principe que le commercial peut ajouter des informations complémentaires avec le bouton "Consulté profil", n'étant pas le coeur de notre sujet: la partie "consulter profil" n'a pas été faite, le bouton existe et on imagine seulement ses éventuelles fonctionnalités.

La présentation des clients a été faite sous forme d'un tableau. On retrouve en premier l'id, car c'est ce qui est le plus important pour le commercial, en effet il devrait rentrer l'id du client lors de la création de son devis. Juste après on retrouve le nom qui permet au commercial d'associer l'id du client à sa personne, car il est évidemment plus simple pour lui de mémoriser un nom qu'un ID, dans cette optique on peut éventuellement envisager une image de profil pour les clients, mais il est rare de retrouver ce concept dans les entreprises. Intervient ensuite les logements d'arrivés et de départ du client, puis l'email. On rappelle qu'à la fin de création du devis, il sera enregistré dans la base de donnée et le client recevra également le devis par mail, il nous est donc indispensable de détenir son email.

## Onglet Devis:

L'onglet devis permet au commercial de remplir toutes les informations nécessaires pour la création du devis de client. La première page sert essentiel aux coûts du transport, il rentre dans un premier temps toutes les informations dans "Informations transports" puis le bouton "Estimation" permet d'établir une estimation des différents coûts. Dans son devis, le client ne recevra que les estimations, il n'a pas besoin de savoir quel véhicule a été choisi, la distance, etc.

Pour le choix du véhicule, on imagine qu'un véhicule est automatiquement sélectionné selon s'il est disponible dans le planning, selon la taille rentrée par le commercial, etc.

The diagram illustrates the 'Onglet Devis' (Quote Tab) form, which is used by a commercial to create a quote for a client. The form is divided into several sections, each with specific input fields and annotations explaining their purpose.

- ID du Client:** A text input field at the top. Annotation: "Le commercial doit rentrer ici l'ID du client pour que l'application sache à quel client correspond le devis".
- Informations transports:** A section containing several input fields:
  - Temps de Transport (en min): Input field. Annotation: "Calculé à partir du temps de transport et le cout du chauffeur".
  - Coût du Chauffeur (€/h): Input field. Annotation: "Calculé à partir du cout du carburant au kilomètres et la distance".
  - Distance de transport (km): Input field. Annotation: "Le commercial rentre les valeurs qu'il souhaite".
  - Coût du Carburant (€/100km): Input field. Annotation: "Le commercial rentre les valeurs qu'il souhaite".
  - Choix du véhicule (en m2): Input field. Annotation: "Le commercial rentre les valeurs qu'il souhaite".
  - Amortissement du véhicule (en €): Input field. Annotation: "Reprend la valeur donnée par le commercial dans «Amortissement du véhicule»".
  - Prix du péage (en €): Input field. Annotation: "Reprend la valeur donnée par le commercial dans «Prix du péage»".
- Estimations:** A section containing four input fields:
  - Coût Chauffeur: Input field.
  - Coût Carburant: Input field.
  - Coût Péage: Input field.
  - Coût Véhicule: Input field.
- Estimation:** A large button to calculate the total cost. Annotation: "Reprend ce qui a été rentré plus haut pour calculer l'estimation des différents coûts et l'afficher".
- Navigation:** Two arrows at the bottom: a left arrow (<) and a right arrow (>).
  - The left arrow is annotated: "Permet de passer à la suite et compléter le devis, autrement: permet de revenir en arrière pour corriger des informations précédemment rentrées (ici la flèche retour n'existe pas puisqu'il s'agit de la première page pour la création du devis)".
  - The right arrow is annotated: "Reprend la valeur donnée par le commercial dans «Amortissement du véhicule»".

La seconde page demande au commercial des informations sur les logements d'arrivés et de départ du client, ce sont des informations qui seront particulièrement utile au déménageurs. Les éventuelles options/services que peut choisir le client sont également cochables ici par le commercial, le coût final variera selon ses services et la simplicité d'un déménagement dans les divers logements.

The screenshot shows a form titled "Services en option:" and "Manutention logement de départ:" and "Manutention logement d'arrivé".

**Services en option:**

- ☐ Démontage et remontage des meubles
- ☐ Emballage des meubles au départ
- ☐ Etat des lieux de sortie
- ☐ Etat des lieux d'entrée
- ☐ Réduction Groupage
- ☐ Devis Tardif

**Manutention logement de départ:**

Volume (en m³):  Nombre d'étages:  Distance Camion / Logement:

☐ Monte Meuble  
☐ Ascenseur

**Manutention logement d'arrivé:**

Volume (en m³):  Nombre d'étages:  Distance Camion / Logement:

☐ Monte Meuble  
☐ Ascenseur

Navigation buttons:

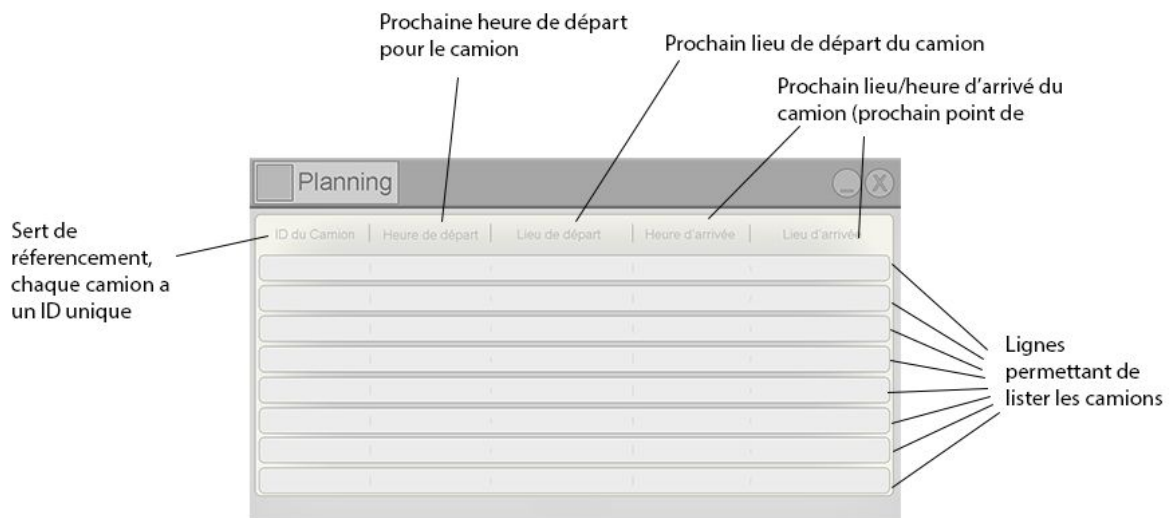
Annotations:

- Le commercial selectionne les options qui ont pu être choisi par le client (points to the "Services en option:" section)
- Le commercial rentre/coche les informations spécifiques au logement d'arrivé et au logement de départ (points to the "Manutention logement de départ:" and "Manutention logement d'arrivé" sections)
- Permet de revenir en arrière dans le devis ou de donner d'autres informations afin de le compléter/finaliser (points to the navigation buttons)

Une fois toutes les informations rentrées par le commercial le devis est créé dans la base de données et un mail est envoyé au client. Sur la page "consulter profil" de l'onglet client (qui pour rappel, n'a pas été faite car le sujet n'est concentré que sur la création du devis) on peut imaginer qu'un bouton "consulter le devis" sera disponible.

### Onglet Planning:

L'onglet planning permet au commercial de voir quel camion est occupé quand et où il se trouve, les camions sont référencés par leur ID qui est unique et nous permet de savoir de quel camion on parle, ensuite on retrouve le même schéma que pour les clients, la présentation est faite sous forme d'un tableau.



### Comment a été créer l'application:

Dans un premier temps, les graphismes de l'application ont été fait à l'aide de photoshop, nous avons fait ce choix car nous apprécions peu l'esthétisme de base proposé par Java, et le fait de tout refaire graphiquement nous laisser une plus grande possibilité dans le positionnement des boutons, des onglets, des zones de textes, etc. Il est d'autant plus simple d'utiliser des images que de refaire un "look and feel". Pour des raisons de simplicité nous avons d'abord fait l'interface graphique puis le code après afin de placer correctement chaque bouton, etc. Il est plus simple de modifier une ligne de code que toute une image.

Pour cette raison, des JLabels ont été utilisés plutôt que des JButton, chaque bouton dispose de 3 états: normal, cliqué et survolé. Chaque bouton existe donc sous ces 3 formes d'images. C'est etats sont contrôlés avec un MouseListener en fonction de la position de la souris. Pour afficher les 3 différents onglets, nous avons représentés cela sous forme de vues avec un JPanel principal incluant un CardLayout avec 3 onglets.

L'onglet devis a une petite spécificité puisqu'il contient lui-même un cardlayout pour ses 2 pages/vues. Dans la même optique, nous n'avons pas utilisé de Layout dans les panneaux dynamique pour pouvoir avoir une image de fond sur laquelle ajouter nos champs de texte (JTextField) et JCheckbox.

Pour la base de données nous avons 3 classes:

- Camion
- Client
- Devis

Chaque classe contient les données spécifiques à une vue.

